Bcfg2 Demo Commands

$ lines describe commands to be run -> describes user input Rest of this is commentary on the actions taken.

1. Initialize the bcfg2 repo

$ sudo bcfg2-admin init -> answer the questions appropriately for your server

After this step is completed, you will have a repository in /var/lib/bcfg2. This repository is a skeleton, containing empty directories for most plugins, and small configuration files for the Metadata plugin, which is required.

2. Start the bcfg2 server $ sudo /etc/init.d/bcfg2-server start

This step starts up the bcfg2 server. You will see a series of messages in syslog like: Dec 26 08:51:19 ubik bcfg2-server[6057]: Bound to port 6789 Dec 26 08:51:36 ubik bcfg2-server[6057]: Processed 121 gamin events in 0.309 seconds. 0 collapsed ...

3. Set the basic group to be public; this allows clients to freely associate themselves with it.

$ sudo vi /var/lib/bcfg2/Metadata/groups.xml -> Change basic group to public='true'

4. Run a client in dry run mode On a client $ sudo /usr/sbin/bcfg2 -q -v -d -n -p basic -x foobat -S https://server:6789

In order, these options will not perform checksum tests, run the client in both verbose and debug modes. -n specifies that Bcfg2 should not make any changes to the client. -p specifies that the client wants to associate itself with the profile group "basic". -x specifies the client password. Finally, -S provides a URL where the server can be contacted.

Notice that the client performs no changes. Also, notice the statistics printed at the end of the run, noting 0 correct entries (none are managed) and a number of unmanaged entries. These entries are probed by the Bcfg2 client and sent to the server. This information can be used later to incrementally manage more of the client's configuration.

-p only needs to be specified once; the server will store the profile group for the client. -x and -S are only needed because there is no bcfg2.conf file on the client (yet). Next, we will create one.

5. Begin to manage /etc/bcfg2.conf

In order to manage a configuration file, two steps must be performed. First, the entry must be included in the list of entries that need to be installed on the client. Then we need to add information about that entry so that it is properly configured on the client system.

5.1 Adding /etc/bcfg2.conf to the list of managed entries for the client.

First, create a new bundle that includes the entry <ConfigFile? name='/etc/bcfg2.conf'/>.

$ cat > /tmp/bcfg2.xml << EOF <Bundle revision='$Revision$' name='bcfg2' version='2.0' >

<ConfigFile? name='/etc/bcfg2.conf'/>

</Bundle> EOF $ sudo mv /tmp/bcfg2.xml /var/lib/bcfg2/Bundler

After this step, we have created a bundle that includes /etc/bcfg2.conf, but no clients will get that bundle.

5.2 Add the bundle bcfg2 to the basic group. $ vi /var/lib/bcfg2/Metadata/groups.xml -> add a client element: <Bundle name='bcfg2'/> to group basic

5.3 Create the literal configuration file to be installed on the client.

$ sudo mkdir -p /var/lib/bcfg2/Cfg/etc/bcfg2.conf

This creates a directory that the Cfg plugin uses to serve data for the config file /etc/bcfg2.conf. Now, we create a file that is the default version of bcfg2.conf that clients should get.

$ cat > /tmp/bcfg2.conf << EOF [communication] protocol = xmlrpc/ssl password = foobat

[components] bcfg2 = https://server:6789 EOF

$ sudo mv /tmp/bcfg2.conf /var/lib/bcfg2/Cfg/etc/bcfg2.conf

6. Run the client again to get the updates to /etc/bcfg2.conf

6.1 Verify the results before installing

$ sudo /usr/sbin/bcfg2 -q -v -d -n -x foobat -S https://server:6789 -> See that it now complains about 1 incorrect entry:

```
ConfigFile /etc/bcfg2.conf does not exist
Failed to read /etc/bcfg2.conf: No such file or directory

Phase: initial
Correct entries:        0
Incorrect entries:      1
Total managed entries:  1
Unmanaged entries:      649

In dryrun mode: suppressing entry installation for:
 ConfigFile:/etc/bcfg2.conf

Phase: final
Correct entries:        0
Incorrect entries:      1
 ConfigFile:/etc/bcfg2.conf
Total managed entries:  1
Unmanaged entries:      649
```

6.2 Run the bcfg2 client in interactive mode

$ sudo /usr/sbin/bcfg2 -q -v -d -I -x foobat -S https://server:6789 -> answer 'y' when it asks if you want to install /etc/bcfg2.conf

Now, it reports one correct entry:

```
Phase: final
Correct entries:        1
Incorrect entries:      0
Total managed entries:  1
```

```
Unmanaged entries:      649
```

Now, we can run bcfg2 without some of the command line cruft: $ sudo /usr/sbin/bcfg2 -q -v -d -n

and we see that everything is copacetic

7. Managing ssh host keys

7.1 Add bundle for ssh

$ cat > /tmp/ssh.xml << EOF <Bundle revision='$Revision$' name='ssh' version='2.0'

origin='https://svn.mcs.anl.gov/repos/bcfg/trunk/repository/Bundler/ssh.xml'>

<ConfigFile? name='/etc/ssh/ssh_host_dsa_key'/> <ConfigFile? name='/etc/ssh/ssh_host_rsa_key'/> <ConfigFile? name='/etc/ssh/ssh_host_dsa_key.pub'/> <ConfigFile? name='/etc/ssh/ssh_host_rsa_key.pub'/> <ConfigFile? name='/etc/ssh/ssh_host_key'/> <ConfigFile? name='/etc/ssh/ssh_host_key.pub'/> <ConfigFile? name='/etc/ssh/ssh_known_hosts'/>

</Bundle>

$ sudo mv /tmp/ssh.xml /var/lib/bcfg2/Bundler

Add ssh bundle into basic group $ sudo vi /var/lib/bcfg2/Metadata/groups.xml -> add <Bundle name='ssh'/> to the basic group

7.2 Validate the repository each time you change an XML file sudo /usr/sbin/bcfg2-repo-validate -v

7.3 Run bcfg2 on the client $ sudo /usr/sbin/bcfg2 -q -v -n

We see incorrect entries for ssh files

```
Phase: initial
Correct entries:        1
Incorrect entries:      7
Total managed entries:  8
Unmanaged entries:      649

In dryrun mode: suppressing entry installation for:
 ConfigFile:/etc/ssh/ssh_host_dsa_key      ConfigFile:/etc/ssh/ssh_host_rsa_key
 ConfigFile:/etc/ssh/ssh_host_dsa_key.pub  ConfigFile:/etc/ssh/ssh_host_rsa_key.pub
 ConfigFile:/etc/ssh/ssh_host_key          ConfigFile:/etc/ssh/ssh_known_hosts
 ConfigFile:/etc/ssh/ssh_host_key.pub

Phase: final
Correct entries:        1
Incorrect entries:      7
 ConfigFile:/etc/ssh/ssh_host_dsa_key      ConfigFile:/etc/ssh/ssh_host_rsa_key
 ConfigFile:/etc/ssh/ssh_host_dsa_key.pub  ConfigFile:/etc/ssh/ssh_host_rsa_key.pub
 ConfigFile:/etc/ssh/ssh_host_key          ConfigFile:/etc/ssh/ssh_known_hosts
 ConfigFile:/etc/ssh/ssh_host_key.pub
Total managed entries:  8
Unmanaged entries:      649
```

7.4 Install client keys into the Bcfg2 repository Now, we pull the ssh host key data for the client out of the uploaded stats and insert it as host-specific copies of these files in /var/lib/bcfg2/SSHBase

$ for key in ssh_host_dsa_key ssh_host_key; do

> sudo bcfg2-admin pull <clientname> ConfigFile? /etc/ssh/$key sudo bcfg2-admin pull <clientname> ConfigFile? /etc/ssh/${key}.pub

done

This for loop pulls data that was collected by the bcfg2 client out of the statistics file and installs it into the repository. This means that the client will keep the same ssh keys and the bcfg2 server can start generating a correct ssh_known_hosts file for the client.

7.5 Run bcfg2 on the client again $ sudo /usr/sbin/bcfg2 -q -v -n

This time, we will only see 1 incorrect entry.

```
Phase: initial
Correct entries:        7
Incorrect entries:      1
Total managed entries:  8
Unmanaged entries:      649

In dryrun mode: suppressing entry installation for:
 ConfigFile:/etc/ssh/ssh_known_hosts

Phase: final
Correct entries:        7
Incorrect entries:      1
 ConfigFile:/etc/ssh/ssh_known_hosts
Total managed entries:  8
Unmanaged entries:      649
```

Now, the only wrong entries is the ssh_known_hosts file!, so let's get it $ sudo /usr/sbin/bcfg2 -q -v -I -> Answer 'y' to the "install /etc/ssh/ssh_known_hosts" question

```
Phase: final
Correct entries:        8
Incorrect entries:      0
Total managed entries:  8
Unmanaged entries:      649
```

7.6 show /etc/ssh/ssh_known_hosts It includes

- all local system keys
- localhost keys (for the local system)
- all systems with keys in bcfg2.

Extra stuff.

- blow away file, or corrupt it, and it gets fixed.
- add a second version of bcfg2.conf for the server (with a host-specific file)